Demo for CLEMM package

August 27, 2018

This a demo for showing how to use CLEMM package

```r
rm(list=ls())
library(CLEMM)

# set parameters for manifold optimization
opts <- c()
opts$record <- 0
opts$mxitr <- 1500
opts$xtol <- 1e-10
opts$gtol <- 1e-10
opts$ftol <- 1e-10

###############################################
# simulate data from given CLEMM model M1-M3 #
###############################################

#K <- 3;r <- 15; u<- 1;N <- 1000;md=1
K <- 4;r <- 15; u<- 2;N <- 1000;md=2
#K <- 4;r <- 15; u<- 3;N <- 1000;md=3
da <- data_generation(r, u, N, K, md)

if (md==1|md==4){pi0 <- c(0.3, 0.2, 0.5)}else {pi0 <- rep(0.25, 4)}


set.seed(1)
idx <- sample(1:K, N, replace = TRUE, prob = pi0)
dat <- matrix(NA, N, r)
for (j in 1:K) {
  if(md <= 3){
    x_tmp <- rmvnorm(sum(idx==j), mean = da$mu[, j], sigma = da$Sigma[, , j])
  }else{ x_tmp <- rmvnorm(sum(idx==j), mean = da$mu[, j], sigma = da$Sigma)}
  dat[idx==j, ] <- x_tmp
}

# bayes classification error rate
bayeserr(dat, pi0, da$mu, da$Sigma, idx)
```

```
## [1] 0.056
```

```r
###########################################
# use true parameters as starting values#
```

```
#########################################
init=list()
init$centers = da$mu
init$wt = pi0
init$cov = da$Sigma

##########################################
# use k-means results as starting values #
##########################################

# init_kmeans <- kmeans(dat, centers=K, nstart=20)
# init=list()
# init$centers = t(init_kmeans$centers)
# init$wt = (init_kmeans$size)/N
# init$cov = array(NA, c(r, r, K))
#
# for(j in 1:K){
#   init$cov[, , j] = cov(dat[init_kmeans$cluster==j,])
# }

#########################################################
# use Hierarchical Clustering results as starting values#
#########################################################

# da_dis <- dist(dat)
# hc_da <- hclust(da_dis, method = "ward.D2")
# sub_grp <- cutree(hc_da, K)
# init = list()
# init$center = matrix(NA, r, K)
# init$wt = matrix(NA, 1, K)
# init$cov = array(NA, c(r, r, K))
# for(j in 1:K) {
#   init$center[, j] <- apply(dat[sub_grp==j, ], 2, mean)
#   init$wt[j] <- sum(sub_grp==j)
#   init$cov[, , j] <- cov(dat[sub_grp==j,])
# }

# GMM estimation and clustering error (CLEMM)
res <- gmm_em(dat, K, iter=800, init=init, typ="G")
gmm_err = clustering_err(K, dat, em_res=res, pi0=pi0, mu=da$mu, Sigma=da$Sigma, idx=idx)

# select envelope dimension
dim_res = env_dim_selection(1:3, dat, K, iter=800, opts=opts, init=init)
# CLEMM estimation and clustering error
```

```
res_clemm = clemm_em(dat, K, u=dim_res$u, iter=800, opts=opts, init=init)
clemm_err = clustering_err(K, dat, em_res=res_clemm, pi0=pi0, mu=da$mu, Sigma=da$Sigma, idx=idx)
```

```
gmm_err
```

```
## $cluster_err
## [1] 0.139
##
## $mean_err
## [1] 1.337468
##
## $wt_err
## [1] 0.2708804
##
## $cov_err
## [1] 2.255976
```

```
clemm_err
```

```
## $cluster_err
## [1] 0.056
##
## $mean_err
## [1] 0.4966855
##
## $wt_err
## [1] 0.08662092
##
## $cov_err
## [1] 0.745007
```

```
##############################################
# simulate data from given CLEMM model M1-M3 #
##############################################
K <- 3;r <- 15; u<- 1;N <- 1000;md=4
#K <- 4;r <- 50; u<- 2;N <- 1000;md=5

da <- data_generation(r, u, N, K, md)
if (md==1|md==4){pi0 <- c(0.3, 0.2, 0.5)}else {pi0 <- rep(0.25, 4)}


set.seed(1)
idx <- sample(1:K, N, replace = TRUE, prob = pi0)
dat <- matrix(NA, N, r)
for (j in 1:K) {
  if(md <= 3){
    x_tmp <- rmvnorm(sum(idx==j), mean = da$mu[, j], sigma = da$Sigma[, , j])
  }else{ x_tmp <- rmvnorm(sum(idx==j), mean = da$mu[, j], sigma = da$Sigma)}
  dat[idx==j, ] <- x_tmp
}

# bayes classification error rate
bayeserr(dat, pi0, da$mu, da$Sigma, idx)

#########################################
# use true parameters as starting values#
#########################################
init=list()
init$centers = da$mu
init$wt = pi0
init$cov = da$Sigma

##########################################
# use k-means results as starting values #
##########################################

# init_kmeans <- kmeans(dat, centers=K, nstart=20)
# init=list()
# init$centers = t(init_kmeans$centers)
# init$wt = (init_kmeans$size)/N
# init$cov = cov(dat)
#


#########################################################
# use Hierarchical Clustering results as starting values#
#########################################################

# da_dis <- dist(dat)
# hc_da <- hclust(da_dis, method = "ward.D2")
# sub_grp <- cutree(hc_da, K)
# init = list()
# init$center = matrix(NA, r, K)
# init$wt = matrix(NA, 1, K)
# init$cov = cov(dat)
# for(j in 1:K) {
#   init$center[, j] <- apply(dat[sub_grp==j, ], 2, mean)
#   init$wt[j] <- sum(sub_grp==j)
# }
# GMM estimation and clustering error (CLEMM)
res <- gmm_em(dat, K, iter=800, init=init, typ="S")
gmm_err = clustering_err(K, dat, em_res=res, pi0=pi0, mu=da$mu, Sigma=da$Sigma, idx=idx)

# select envelope dimension
dim_res = env_dim_selection(1:10, dat, K, iter=800, opts=opts, init=init, typ="S")

# CLEMM estimation and clustering error
res_clemm = clemm_em(dat, K, u=dim_res$u, iter=800, opts=opts, init=init, typ="S")
clemm_err = clustering_err(K, dat, em_res=res_clemm, pi0=pi0, mu=da$mu, Sigma=da$Sigma, idx=idx)
```

```
gmm_err
```

```
## $cluster_err
## [1] 0.076
##
## $mean_err
## [1] 0.1353736
##
## $wt_err
## [1] 0.05668384
##
## $cov_err
## [1] 0.01113369
```

```
clemm_err
```

```
## $cluster_err
## [1] 0.069
##
## $mean_err
## [1] 0.0970314
##
## $wt_err
## [1] 0.05201009
##
## $cov_err
## [1] 0.007966649
```